

Passwortklasse

Themen

Mit dieser Aufgabe wollen wir Folgendes trainieren:

- Objektorientierung
- Klassen, Instanzvariablen, Methoden

Beschreibung

Wir wollen eine Klasse mit Namen `Password` implementieren, die ein Passwort repräsentiert.

Intern wird das Passwort als `char`-Array verwaltet. Folgende Methoden sollen wir implementieren:

- Der Konstruktor bekommt das Passwort als `char`-Array übergeben und setzt die strikte Kapselung durch. Zudem wird dafür gesorgt, dass das übergebene Passwort nur noch aus Leerzeichen besteht.
- Die Java-Funktion `isStrong()` aus Aufgabe 16 ist als Klassenmethode enthalten. Es reicht, wenn wir die Signatur der Methode angeben.
- Mit der Methode `changePassword()`, kann das Passwort geändert werden. Hierzu muss sowohl das „alte“ Passwort als auch das „neue“ Passwort übergeben werden. Stimmt das übergebene „alte“ Passwort nicht, wird nichts geändert, und die Methode liefert *false* zurück. Handelt es sich beim „neuen“ Passwort nicht um ein starkes Passwort, ändert sich ebenfalls nichts, und die Methode liefert gleichfalls *false* zurück. Wie beim Konstruktor soll auch diese Methode die strikte Kapselung durchsetzen.
- Die Methode `deletePassword()` löscht das Passwort unwiderruflich aus dem Speicher.

Aufgabenstellung

Implementiere die oben beschriebene Klasse `Password`.

Testfälle

Dieser Testcode sollte zweimal den Wert *false* und einmal den Wert *true* im Terminal zurückgeben, wenn er in der `setup()`-Funktion ausgeführt wird:

```
char[] pwd_first = "PassWD15!!".toCharArray();
char[] pwd_weak = "1234567890".toCharArray();
char[] pwd_strong = "NewPWD16!!".toCharArray();
// Erzeuge ein neues Passwortobjekt und setze das
// Passwort auf "PassW15!!"
Password pwd = new Password(pwd_first);
// Ändere Passwort auf "1234567890"
// Klappt nicht, weil das neue Passwort schwach ist
System.out.println(pwd.change(pwd_first, pwd_weak));
// Verändern der lokalen pwd_first-Variablen
```

```
// Darf keine Auswirkungen auf das Passwort im pwd-Objekt haben.  
pwd_first[0] = 'p';  
// Ändere Passwort auf "NewPWD16!!"  
// Klappt nicht, weil das alte Passwort nicht stimmt  
System.out.println(pwd.change(pwd_first, pwd_strong));  
// Zurücksetzen der lokalen pwd_first-Variablen  
// Darf keine Auswirkungen auf das Passwort im pwd-Objekt haben.  
pwd_first[0] = 'P';  
// Ändere Passwort auf "NewPWD16!!"  
// Klappt, weil das alte Passwort stimmt und  
// das neue Passwort stark ist  
System.out.println(pwd.change(pwd_first, pwd_strong));
```

Algorithmische Tipps

Wenn du stockst und nicht weiterweissst, dann versuch mal Folgendes:

- Die Aufgabe zum starken Passwort ergibt eine Funktion, die du hier einbauen kannst.
- Der char-Array ist ein Referenzdatentyp und muss kopiert werden.
- Ebenso musst du beim Passwortvergleich jeweils Zeichen für Zeichen vergleichen.